1st International Conference "From Scientific Computing to Computational Engineering
1st IC-SCCE
Athens, 8-10 September, 2004

A NON-MONOTONE DIMENSION-REDUCING CONIC METHOD FOR UNCONSTRAINED OPTIMIZATION

G. E. Manoussakis, D. G. Sotiropoulos, T. N. Grapsa, and C. A. Botsaris

Department of Mathematics, University of Patras, GR-261.10 Patras, Greece
E-mail: {gemini, dgs, grapsa}@math.upatras.gr, botsaris@otenet.gr

Keywords: Unconstrained Optimization, Conic model, Dimension-Reducing Method, Barzilai-Borwein step length, Non-monotone.

Abstract. *In a recent article, we introduced a method based on a conic model for unconstrained optimization. The acceleration of the convergence of this method was obtained by choosing more appropriate points in order to apply the conic model. In particular, we applied in the gradient of the objective function a dimension-reducing method for the numerical solution of a system of algebraic equations. In this work, we incorporate in the previous method the non-monotone Armijo line search, introduced by Grippo, Lampariello and Lucidi, combined with the Barzilai and Borwein steplength, in order to further accelerate the convergence. The new method does not guarantee descent in the objective function value at each iteration. Nevertheless, the use of this non-monotone line search allows the objective function to increase at some iteration without affecting the global convergence properties. The new method has been implemented and tested in well known test functions. It converges in n+1 iterations on conic functions and, as numerical results indicate, rapidly minimizes general functions.*

1. INTRODUCTION

We deal with the general unconstrained minimization problem:

$$\min f(x), \quad x \in \mathbb{R}^n$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice-continuously differentiable function in n variables $x = (x_1, x_2, \dots x_n)$.

Most methods for unconstrained optimization are based on a quadratic model. Various authors have introduced non-quadratic algorithms [2, 4, 13]. Davidon in [4] introduced a conic model for unconstrained optimization. The conic model function has the form:

$$c(x) = \frac{1}{2}\frac{x^T Q x}{\left(1 + p^T x\right)^2} + \frac{b^T x}{1 + p^T x} + a, \qquad (1.1)$$

where Q is a $n \times n$ symmetric matrix and $p \in \mathbb{R}^n$ is the vector defining the horizon of the conic function, i.e. the hyper plane where $c(x)$ takes an infinite value, which is defined by the equation $1 + p^T x = 0$.

Botsaris and Bacopoulos in [2] presented a conic model of the form

$$c(x) = \frac{1}{2}\frac{1 + p^T x}{1 + p^T \beta}\nabla c(x)(x - \beta) + c(\beta),$$

where β is the location of the minimum. As can be easily shown this form of the conic function is equivalent to (1.1). The conic model in this form does not involve and, therefore, it does not require, an estimate of the conjugacy matrix Q or the Hessian matrix of the objective function.

The conic function $c(x)$ has a unique minimum whenever the symmetric matrix Q is positive definite. The location of the minimizer is determined by solving the equation:

$$\frac{\beta}{1 + p^T \beta} + Q^{-1}\beta = 0 \Leftrightarrow \beta = -\frac{Q^{-1}\beta}{1 + p^T Q^{-1}\beta},$$

provided that such a solution exists, i.e. $1 + p^T Q^{-1}\beta \neq 0$.

In [3] a gradient method has been proposed, where the search direction is always the negative gradient direction, but the choice of the step length is not the classical choice of the steepest descent method. The motivation for this choice is that it provides two-point approximation to the secant equation underlying quasi-Newton methods [23]. This yields the iteration:

$$x_{i+1} = x_i - t_i g\left(x_i\right),\ i = 0,1,2,\ldots$$

where $g_i = g\left(x_i\right)$ is the gradient of the objective function at x_i and the step t_i is given by:

$$t_{i+1} = \frac{\left(x_{i+1} - x_i\right)^T \left(x_{i+1} - x_i\right)}{\left(x_{i+1} - x_i\right)^T \left(g_{i+1} - g_i\right)}.$$

The choice of the step length is related to the eigenvalues of the Hessian at the minimizer and not to the function value.

In [12] a non-monotone line search for Newton-type methods has been proposed and in [22], [24] some computational advantages of this technique have been pointed out. The method imposes that the objective function value f at each iteration must satisfy the Armijo's condition with respect to the maximum objective function value of a prefixed number M of previous iterations. Formally:

$$f\left(x_k - t_k \nabla f\left(x_k\right)\right) - \max_{0 \le j \le M}\left\{f\left(x_{k-j}\right)\right\} \le -\sigma t_k \left\|\nabla f\left(x_k\right)\right\|^2$$

where M is a nonnegative integer, and $0 < \sigma < 1$. The above condition allows an increase in the function values without affecting the global convergence properties [12, 16]. This method has low storage requirements and inexpensive computations.

In [17] we improved the convergence of the conic method presented in [2], substituting the classical Armijo's line search and step with the above non-monotone line search and the Barzilai and Borwein step respectively.

In [11] Grapsa and Vrahatis proposed a dimension-reducing method for unconstrained optimization (the DROPT method). This method incorporates the advantages of Newton and SOR algorithms. In particular, although it uses reduction to simpler one-dimensional nonlinear equations, it generates a sequence in $\mathbb{R}^{n-1}$ which converges quadratically to the $n-1$ components of the optimum while the remaining component is evaluated separately using the final approximations of the others. For this component an initial guess is not necessary and it is at the user's disposal to choose which will be the remaining component, according to the problem. Also the DROPT method does not directly need any gradient evaluation.

In [18] we have used the DR-method to obtain more appropriate points to apply the conic model. In this way we have accelerated more the convergence of the conic method.

In this paper we use the DR-Method to obtain more appropriate points to apply the conic model in combination with the Barzilai and Borwein step. We also incorporate the non-monotone philosophy, applying the acceptance criterion for each iteration with respect of the maximum objective function value of M previous iterations.

2. THE NEW CONIC METHOD

If we assume that the function f to be minimized is conic, then the following equation is satisfied:

$$\left[2f\left(x\right)p^T + \left(1 + p^T x\right)g^T\left(x\right)\right]\beta - 2\left(1 + p^T\beta\right)f\left(\beta\right) = \left(1 + p^T x\right)g^T\left(x\right)x - 2f\left(x\right). \qquad (2.1)$$

If the horizon p is known, then by calculating (2.1) at $n+1$ distinct points $x_1, x_2, \ldots, x_{n+1}$ we have a $\left(n+1\right)\times\left(n+1\right)$ linear system:

$$A\alpha = s - \phi, \qquad \alpha = \begin{bmatrix}\beta \\ \omega\end{bmatrix} \qquad (2.2)$$

where

$$A = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n+1} \end{pmatrix}\begin{bmatrix} p^T & 0 \end{bmatrix} + G \qquad G = \begin{pmatrix} \left(1 + p^T x_1\right) g_1^T & -1 \\ \left(1 + p^T x_2\right) g_2^T & -1 \\ \vdots & \vdots \\ \left(1 + p^T x_{n+1}\right) g_{n+1}^T & -1 \end{pmatrix}$$

$$s=\begin{pmatrix}\left(1+p^Tx_1\right)g_1^Tx_1\\ \left(1+p^Tx_2\right)g_2^Tx_2\\ \vdots\\ \left(1+p^Tx_{n+1}\right)g_{n+1}^Tx_{n+1}\end{pmatrix}\qquad \phi=2\begin{pmatrix}f_1\\ f_2\\ \vdots\\ f_{n+1}\end{pmatrix}\qquad \omega=2\left(1+p^T\beta\right)f\left(\beta\right)$$

The $n+1$ distinct points $x_1,x_2,\dots,x_{n+1}$ needed for the formulation of the system (2.2), are found with the help of the DROPT-method. This way we obtain a better set of points, so the convergence of the conic algorithm is accelerated.

According to the DROPT method, to obtain a sequence $\left\{x^k\right\}, k=0,1,\dots$ of points in $\mathbb{R}^n$ which converges to a local optimum point $x^*=\left(x_1^*,x_2^*,\dots,x_n^*\right)$ of the function f, we consider the sets $B_i, i=1,2,\dots,n$ to be the connected component of $g_i^{-1}\left(0\right)$ containing x^* on which $\partial_n g_i\neq 0$, for $i=1,2,\dots,n$ respectively, where $g\left(x\right)=\left(g_1\left(x\right),g_2\left(x\right),\dots,g_n\left(x\right)\right)^T=\nabla f$ the gradient of the objective function f. Applying the Implicit Function Theorem [5, 7, 20], for each one of the components $g_i, i=1,2,\dots,n$ we can find open neighborhoods $A_1^*\subset\mathbb{R}^{n-1}$ and $A_{2,i}^*\subset\mathbb{R}, i=1,2,\dots n$ of the points $w^*=\left(x_1^*,x_2^*,\dots,x_{n-1}^*\right)$ and x_n^* respectively, such that for any $w=\left(x_1,x_2,\dots,x_{n-1}\right)\in\bar{A}_1^*$ there exist unique mappings ϕ_i defined and continuous in A_1^* such that:

$$x_n=\varphi_i\left(w\right)\in\bar{A}_{2,i}^*,\ i=1,\dots,n$$

and

$$g_i\left(w,\varphi_i\left(w\right)\right)=0,\ i=1,\dots,n$$

Moreover, the partial derivatives $\partial_j\varphi_i=\dfrac{\partial\varphi_i}{\partial x_j}$, $j=1,\dots,n-1$ exist in $\bar{A}_1^*$ for each $\varphi_i, i=1,\dots,n$, they are continuous in $\bar{A}_1^*$ and they are given by:

$$\partial_j\varphi_i\left(w\right)=-\frac{\partial_j g_i\left(w;\varphi_i\left(w\right)\right)}{\partial_n g_i\left(w;\varphi_i\left(w\right)\right)},\qquad i=1,\dots,n,\qquad j=1,\dots,n-1$$

Next, we utilize Taylor's formula to expand the $\varphi_i\left(w\right)$, $i=1,\dots,n$, about w^p. By straightforward calculations, we can obtain the following iterative scheme for the computation of the $n-1$ components of x^*:

$$w^{p+1}=w^p+A_p^{-1}V_p,\qquad p=0,1,\dots$$

where:

$$w^p=\left[x_i^p\right],\qquad i=1,\dots,n-1$$

$$A_p=\left[\frac{\partial_j g_i\left(w^p;x_n^{p,i}\right)}{\partial_n g_i\left(w^p;x_n^{p,i}\right)}-\frac{\partial_j g_n\left(w^p;x_n^{p,n}\right)}{\partial_n g_n\left(w^p;x_n^{p,n}\right)}\right],\ i,j=1,\dots,n-1\qquad (2.3)$$

$$V_p=\left[v_i\right]=\left[x_n^{p,i}-x_n^{p,n}\right],\qquad i=1,\dots,n-1\qquad (2.4)$$

with $x_n^{p,i}=\varphi_i\left(w^p\right)$. After a desired number of iterations, say $p=m$, the nth component of x^* can be approximated by means of the following relation:

$$x_n^{m+1}=x_n^{m,n}-\sum_{j=1}^{n-1}\left\{\left(x_j^{m+1}-x_j^m\right)\frac{\partial_j g_n\left(w^m;x_n^{m,n}\right)}{\partial_n g_n\left(w^m;x_n^{m,n}\right)}\right\}\qquad (2.5)$$

Remark 1 Relative procedures for obtaining x^* can be constructed by replacing x_n with any one of the components $x_1,\dots,x_{n-1}$, for example x_{int}, and taking $w=(x_1,\dots,x_{\text{int}-1},x_{\text{int}+1},\dots,x_n)$.

Remark 2 The above described method does not require the expressions φ_i but only the values $x_n^{p,i}$ which are given by the solution of the one-dimensional equations $g_i\left(x_1^p,\dots,x_{n-1}^p,\cdot\right)=0$. So, by holding $w^p=\left(x_1^p,\dots,x_{n-1}^p\right)$ fixed, we can solve the equations $g_i\left(w^p;r_i^p\right)=0$, $i=1,\dots,n$ for r_i^p in the interval (a,b) with accuracy δ.

Of course we can use any other one-dimensional method to solve the above equations. Here we employ a modified bisection method [11, 28, 30]. The only computable information required by this bisection method is the algebraic signs of the function g_i. Moreover, it is the only method that can be applied to problems with imprecise function values.

After the system (2.2) is formulated, using the DR points, we proceed with the conic method. So, let

$$\rho_i=\frac{1+p^Tx_{i+1}}{1+p^Tx_i}=\frac{\Delta f+k_i}{g_{i+1}^T\Delta x}$$

where $\Delta f=f_{i+1}-f_i$, $\Delta x=x_{i+1}-x_i$, and $\left[\Delta f^2-g_{i+1}^T\Delta x g_i^T\Delta x\right]^{\frac{1}{2}}$ provided that the quantity under the square root is non-negative. If this quantity is negative then the conic method cannot proceed. In this case, using DR Method, we get a new point x, evaluate a new equation (2.1) and we restart the conic procedure to solve the modified system (2.2).

The gauge vector p can be determined by solving the linear system:

$$Zp=r \qquad (2.6)$$

where $z_i=x_{i+1}-\rho_i x_i$, $Z=\begin{bmatrix} z_1\\ z_2\\ \vdots\\ z_n\end{bmatrix}$, $\eta_i=\rho_i-1$, $r=\begin{bmatrix}\eta_1\\ \eta_2\\ \vdots\\ \eta_n\end{bmatrix}$.

From (2.2), (2.6), we have that the location of the minimum β can be determined through the equation:

$$p=Z^{-1}r,\ \ \alpha=A^{-1}(s-\phi)$$

We carry out the necessary inversions recursively as new points are constructed by the algorithm. Using Householder's formula for matrix inversion it can be verified that:

$$Z_i^{-1}=Z_{i-1}^{-1}-\frac{Z_{i-1}^{-1}e_l\left(z_i^TZ_{i-1}^{-1}-e_l^T\right)}{z_i^TZ_{i-1}^{-1}e_l} \qquad (2.7)$$

and

$$p_i=p_{i-1}+\frac{Z_{i-1}^{-1}e_l\left(\eta_i-z_i^Tp_{i-1}\right)}{z_i^TZ_{i-1}^{-1}e_l} \qquad (2.8)$$

provided that $\left|z_i^TZ_{i-1}^Te_l\right|$ is bounded away from zero. We note that e_l is a vector with zero elements except the position $l=i$, where it has unity.

The solution of the linear system is proved to be:

$$\alpha=u-\frac{1+q^Tu}{1+q^Tv}\cdot v, \qquad (2.9)$$

where $q=\begin{bmatrix}p\\0\end{bmatrix}$, $u=G^{-1}s$ and $v=G^{-1}\phi$.

Let us further define:

$$\Lambda_i = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_i \end{bmatrix}, \qquad \lambda_i = \frac{1+p_i^T x_i}{1+p_{i-1}^T x_i} \quad \text{and} \qquad y_{i+1}^T = \left[\frac{1+p_i^T x_{i+1}}{\lambda_i} g_{i+1}^T \quad -1 \right] \qquad (2.10).$$

Then G_{i+1}^{-1} can be computed according to the recursive formula:

$$G_{i+1}^{-1} = \frac{\Lambda_i}{\lambda_i}\left[G_i^{-1} - \frac{G_i^{-1} e_j \left(y_{i+1}^T G_i^{-1} - e_j^T \right)}{y_{i+1}^T G_i^{-1} e_j} \right] \qquad (2.11)$$

provided that $\left| y_{i+1}^T G_i^{-1} e_c \right|$ is bounded away from zero.

The recursive equations for the vectors u and v, required to compute α_{i+1} from (2.9), are found to be:

$$u_{i+1} = \Lambda_i \left[u_i - \frac{G_i^{-1} e_j \left(\theta_{i+1} - y_{i+1}^T u_i \right)}{y_{i+1}^T G_i^{-1} e_j} \right] \quad \text{and} \quad v_{i+1} = \frac{\Lambda_i}{\lambda_\iota} \left[v_i - \frac{G_i^{-1} e_j \left(\xi_{i+1} - y_{i+1}^T v_i \right)}{y_{i+1}^T G_i^{-1} e_j} \right] \qquad (2.12)$$

where

$$\theta_{i+1} = \frac{1+p_i^T x_{i+1}}{\lambda_i} g_{i+1}^T x_{i+1} s \quad \text{and} \quad \xi_{i+1} = 2 f_{i+1} \quad (2.13)$$

The proposed method is illustrated in the following algorithms in pseudo-code where x^0 indicates the starting point, $a = (a_1, a_2, \ldots, a_n)$, $b = (b_1, b_2, \ldots, b_n)$ indicate the endpoints in each coordinate direction which are used for the above mentioned one-dimensional bisection method, with predetermined accuracy δ, MIT the maximum number of iterations required and $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ the predetermined desired accuracies.

Procedure DROPT **(Dimension-Reducing Optimization)** [11].

Step 1. Input x^0; a; b; δ; MIT; ε_1; ε_2

Step 2. Set $p = -1$.

Step 3. If $p < MIT$ replace p by $p+1$ and go to next step; otherwise, go to Step 14.

Step 4. If $\left\| g\left(x^p\right) \right\| < \varepsilon_1$ go to Step 14.

Step 5. Find a coordinate int such that the following relation holds:

$$\operatorname{sgn} g_i \left(x_1^p, \ldots, x_{\text{int}-1}^p, a_{\text{int}}, x_{\text{int}+1}^p, \ldots, x_n^p \right) \cdot \operatorname{sgn} g_i \left(x_1^p, \ldots, x_{\text{int}-1}^p, b_{\text{int}}, x_{\text{int}+1}^p, \ldots, x_n^p \right) = -1,$$

for all $i = 1, 2, \ldots, n$. If this is impossible, apply Armijo's method and go to Step 4.

Step 6. Compute the approximate solutions r_i, $i = 1, 2, \ldots, n$ of the equation

$\operatorname{sgn} g_i \left(x_1^p, \ldots, x_{\text{int}-1}^p, r_i, x_{\text{int}+1}^p, \ldots, x_n^p \right) = 0$ by applying the modified bisection in $\left(a_{\text{int}}, b_{\text{int}}\right)$ within accuracy δ. Set $x_{\text{int}}^{p,i} = r_i$.

Step 7. Set $y^p = \left(x_1^p, \ldots, x_{\text{int}-1}^p, x_{\text{int}+1}^p, \ldots, x_n^p \right)$.

Step 8. Set the elements of the matrix A_p of Relation (2.3) using x_{int} instead of x_n.

Step 9. Set the elements of the vector V_p Relation (2.4) using x_{int} instead of x_n.

Step 10. Solve the $(n-1)\times(n-1)$ linear system $A_p s^p = -V_p$ for s^p.

Step 11. Set $y^{p+1} = y^p + s^p$.

Step 12. Compute x_{int} by virtue of Relation (2.5) and set $x^p = \left(y^p ; x_{\text{int}} \right)$.

Step 13. If $s^p \le \varepsilon_2$ go to Step 14; otherwise return to Step 3.

Step 14. Output $\{ x^p \}$.

The criterion in Step 5 ensures the existence of the solution r_i which will be computed at Step 6. If this criterion is not satisfied we apply Armijo's method [1, 11, 28] for a few steps and then try again with DR method. Our experience is that in many examples studied in various dimensions as well as for all the problems studied in this paper (see below in Numerical Applications) the application of such a subprocedure is not necessary. We have merged it in our algorithm for completeness.

Main Algorithm: **Non Monotone Dimension Reducing Conic Method.**

Step 1. Assume x^0 is given. Set $i=0$; $alpha=0.1$; $k=0$; $j=0$; $W(j)=f_0$.

Step 2. Set $d_0=-g_0$.

Step 3. Use DR-Method to get a point x_1.

Step 4. Set $\alpha_0^T=\begin{bmatrix} x_1^T & 0\end{bmatrix}=u_0^T$, $G_0=Z_0=I$, $v_0=p_0=0$, $\lambda_0=jc=lc=1$.

Step 5. If $\|g_0\|\le\varepsilon_1\left(1+|f_0|\right)$ then stop; else go to Step 6.

Step 6. Use (2.10) to calculate L_{i+1}, y_{i+1}.

Step 7. If $\left|y_{i+1}^T G_i^{-1} e_c\right|<\varepsilon_3$, then set $x_0=x_{i+1}$ and go to Step 3; else go to Step 8.

Step 8. Use (2.9), (2.11), (2.12), (2.13) to calculate a_{i+1}.

Step 9. If $\left|\left(x_i-\beta_i\right)^T g_i\right|<\varepsilon_4$ then set $x_0=x_{i+1}$ and go to Step 3; else go to Step 10.

Step 10. If $f\left(\beta_{i+1}\right)\le f_{\max}$, then store the function value in the record, set $x_{i+1}=\beta_{i+1}$ and go to step 11; else go to step 3.

Step 11. Set $i=i+1$; If $jc=n+1$ then reset $jc=1$, else $jc=jc+1$.

Step 12. Set $d_i=\mu_i\left(x_i-\beta_i\right)$ where $\mu_i=-sign\left\{g_i^T\left(x_i-\beta_i\right)\right\}$.

Step 13. If $\|d_i\|+|\gamma_i|\le M$, then determine the Barzilai step t_i, set $x_{i+1}=x_i+t_i d_i$ and go to step 14; else set $x_0=x_i$ and go to Step 3.

Step 14. If $\delta_i=\left(f_{i+1}-f_i\right)^2-g_{i+1}^T\left(x_{i+1}-x_i\right)g_i^T\left(x_{i+1}-x_i\right)<0$, then using DR-Method find an x_{i+1}, and repeat this procedure until the new x_{i+1} so obtained satisfies $\delta_i>0$; go to Step 15.

Step 15. If $\left|z_i^T Z_{i-1} e_l\right|<\varepsilon_3$, then set $x_0=x_{i+1}$ and go to Step 3; else go to Step 16.

Step 16. Use (2.7), (2.8) to calculate Z_{i+1}^{-1}, p_{i+1}

Step 17. If $lc=n$, then reset $lc=1$; else set $lc=lc+1$.

Step 18. Go to Step 5.

3. NUMERICAL APPLICATIONS

The new method has been implemented using a new FORTRAN90 program named SCONIC. SCONIC has been tested on a Pentium IV PC compatible with random problems of various dimensions. Our experience is that the algorithm behaves predictably and reliably and the results have been quite satisfactory.

Next we compare the numerical results obtained, for various starting points, by applying algorithms (Armijo's quadratic method [1], Fletcher-Reeves [6], Polak-Ribiere [23]), including the classic conic method, with the corresponding results of our method.

For the following problems, the reported parameters are:

- $x_0=\left(x_1,x_2,\ldots,x_n\right)$: the starting point,
- $x^*=\left(x_1^*,x_2^*,\ldots,x_n^*\right)$: approximate local minimum,
- IT: the total number of iterations required in obtaining x^*,
- FE: the total number of function and gradient evaluations,
- AS: the total number of algebraic signs needed by DROPT.

The index α indicates the classical starting point and D indicates divergence or non convergence after 10000 iterations. The approximate local optimum x^*, as long as all the function values were computed within an accuracy of $\varepsilon = 10^{-15}$. We set the size of the line search record to be $M = n$.

Example 1: Extended Rosenbrock Function [19]

$$f(x) = \left[10\left(x_2 - x_1^2\right)\right]^2 + \left(1 - x_1\right)^2 \text{ with } f\left(x^*\right) = 0 \text{ at } x^* = (1,1)$$

	Armijo		FR		PR		CONIC		SCONIC		
x_0	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	AS
(-1.2, 1)*	1881	21396	142	2545	19	364	20	97	6	24	80
(-3, 6)	5960	74560	194	4462	23	455	113	602	5	20	60
(-2, 2)	1828	20852	29	480	15	290	D		5	20	60
(-3, 3)	5993	74364	130	2939	26	509	172	830	5	20	60
(1, 20)	D		259	5732	32	689	D		1	6	20
(10, 10)	D		310	7469	26	526	D		5	20	60
(100, 100)	D		D		33	746	D		3	16	60
(-2000, 2000)	2542	35743	D		93	2466	D		5	20	60
(-1.2, -1)	745	8283	39	615	15	278	80	341	5	23	80
(0, -1.2)	778	8697	31	469	17	315	41	181	7	21	40
(3, 3)	1325	15824	134	2992	26	509	D		12	37	80
(10, -10)	782	8750	31	511	16	308	64	278	5	17	80

Example 2: Freudenstein and Roth Function [19]

$$f(x) = \left[-13 + x_1 + \left[\left(5 - x_2\right)x_2 - 2\right]x_2\right]^2 + \left[-29 + x_1 + \left[\left(x_2 + 1\right)x_2 - 14\right]x_2\right]^2 \text{ with } f\left(x^*\right) = 0 \text{ at}$$
$$x^* = (5,4) \text{ and } f\left(x^*\right) = 48.9842 \text{ at } x^* = (11.41\ldots, -0.8968\ldots)$$

	Armijo		FR		PR		CONIC		SCONIC		
x_0	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	AS
(0.5, -2)*	1827	24155	18	356	8	187	107	501	51	257	1020
(0.5, 1000)	1380	18770	D		D		58	272	31	120	380
(-2, -2)	1119	14625	19	336	8	180	14	70	51	257	1020
(-20, 20)	1851	24986	24	451	10	211	136	630	19	60	140
(4.5, 4.5)	1239	16289	18	342	9	196	10	36	8	25	60
(10, 100)	1845	24664	10	200	9	194	103	459	24	85	240
(12, 2)	2027	26886	70	1145	8	130	11	30	52	261	1040
(4, -1000)	1886	25597	D		D		33	136	72	362	1440

Example 3: Brown badly scaled Function [19]

$$f(x) = \left(x_1 - 10^6\right)^2 + \left(x_2 - 2\cdot 10^{-6}\right)^2 + \left(x_1 x_2 - 2\right)^2 \text{ with } f\left(x^*\right) = 0 \text{ at } x^* = \left(10^6, 2\cdot 10^{-6}\right)$$

	Armijo		FR		PR		CONIC		SCONIC		
x_0	IT	FE	IT	FE	IT	FE	IT	FE	IT	FE	AS
(1, 1)*	D		D		D		12	67	5	20	40
(-1, 1)	D		D		D		11	65	13	43	120
(2, 2)	D		D		D		12	68	16	49	120
(10000, 1)	D		D		D		7	54	10	32	80
(-1000, 1000)	D		D		D		17	93	15	57	180
(10000000, 1)	D		D		D		6	60	8	25	60

REFERENCES

[1] Armijo, L. (1966), "Minimization of functions having Lipschitz continuous first partial derivatives", Pacific J. Math., 16, 1–3.
[2] Bacopoulos A. and Botsaris, C.A. (1992), "A new conic method for unconstrained minimization", J. Math. Anal. Appl., 167, 12–31.
[3] Barzilai, J. and Borwein, J.M. (1988), "Two-point step size gradient methods", IMA J. Num. Anal., 8, 141–148.
[4] Davidon, W.C. (1959), Variable metric methods for minimization, A. E. C., Research and Development Report, No. ANL–5990, Argonne Nat'l Lab., Argonne, Illinois.
[5] J. Dieudonne (1969), *Foundations of modern analysis*, Academic Press, New York.
[6] Fletcher, R. and Reeves, C. (1964), "Function minimization by conjugate gradients", Comput. J., 7, 149–154.
[7] T.N. Grapsa and M.N. Vrahatis (1989), "The implicit function theorem for solving systems of nonlinear equations in R^2", Inter. J. Computer Math. 28 171–181.
[8] T.N. Grapsa and M.N. Vrahatis (1990), "A dimension–reducing method for solving systems of nonlinear equations in R^n , Inter. J. Computer Math. 32 205–216.
[9] T.N. Grapsa, M.N. Vrahatis and T.C. Bountis (1990), "Solving systems of nonlinear equations in R^n using a rotating hyperplane in R^{n+1}", Inter. J. Computer Math. 35 133–151.
[10] T.N. Grapsa and M.N. Vrahatis (1995), "A new dimension–reducing method for solving systems of nonlinear equations", Inter. J. Computer Math. 55 235–244.
[11] T.N. Grapsa and M.N. Vrahatis (1996), "A dimension–reducing method for unconstrained optimization", Journal ofComputational and Applied Mathematics 66 239–253.
[12] Grippo, L., Lampariello, F. and Lycidi, S. (1986), "A nonmonotone line search technique for Newton's method", SIAM J. Numer. Anal., 23, 707–716.
[13] Jacobson, D.H. and Oksman, W.,(1972), "An algorithm that minimizes homogeneous functions of n variables in n + 2 iterations and rapidly minimizes general functions", J. Math. Anal. Appl. 38, 535–552
[14] B. Kearfott (1979), "An efficient degree–computation method for a generalized method of bisection", Numer. Math. 32, 109–127.
[15] R. B. Kearfott (1987), "Some tests of generalized bisection", ACM Trans. Math. Software 13, 197–220.
[16] M. Kupferschmid and J.G. Ecker (1987), "A note on solution of nonlinear programming problems with imprecise function and gradient values", Math. Program. Study 31 129–138.
[17] Manoussakis, G.E., Sotiropoulos, D.G., Botsaris, C.A., and Grapsa, T.N. (2002), *A Non–Monotone Conic Method for Unconstrained Optimization, In: Proceedings of 4th GRACM, Congress on Computational Mechanics*, 27-29 June, University of Patras, Greece.
[18] Manoussakis, G. E., Grapsa, T. N., Bosaris, C. A. (2003), *A Dimension - Reducing Conic Method for Unconstrained Optimization, In Proceedings of HERCMA 2003*, September 2003, Athens
[19] More, B.J., Garbow, B.S. and Hillstrom, K.E. (1981), "Testing unconstrained optimization", ACM Trans. Math. Software, 7, 17–41.
[20] J.M. Ortega and W.C. Rheinbolt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.
[21] A. Ostrowski (1973), *Solution of equations in Euclidean and Banach spaces*, Third Edition, Academic Press, London.
[22] Plagianakos, V.P., Sotiropoulos, D.G. and Vrahatis, M.N. (1998), "A Nonmonotone Backpropagation Training Method for Neural Networks", Dept. of Mathematics, Univ. of Patras, Technical Report No.98-04.
[23] Polak, E. (1971), *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York.
[24] Raydan, M. (1997), "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem", SIAM J. Optim., 7, 26–33.
[25] W. C. Rheinboldt (1974), *Methods for solving systems of equations*, SIAM, Philadelphia.
[26] J. F. Traub (1964), *Iterative methods for the solution of equations*, Prentice–Hall., Inc., Englewood Cliffs, NJ.
[27] M. N. Vrahatis (1988), "CHABIS: A mathematical software package for locating and evaluating roots of systems of nonlinear equations", ACM Trans. Math. Software 14, 330–336.
[28] M.N. Vrahatis, G.S. Androulakis and G.E. Manoussakis (1996), "A new unconstrained optimization method for imprecise function and gradient values", J. Math. Anal. Appl., 197, 586-607.
[29] M. N. Vrahatis and K. I. Iordanidis (1986), "A rapid generalized method of bisection for solving systems of non–linear equations", Numer. Math. 49, 123–138.